INTEIRO: qualquer número inteiro negativo, nulo ou positivo. Ex.: −5, 0, 235.

REAL: qualquer número real negativo, nulo ou positivo. Ex.: −5, 30.5, 0, 40.

CARACTER: qualquer conjunto de caracteres alfanuméricos. Ex.: "AB", "XYZ", "ABACATE"

LÓGICO: conjunto de valores *FALSO* ou *VERDADEIRO* em proposições lógicas.

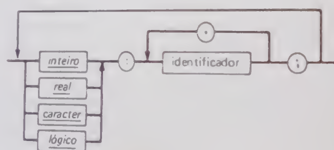
2.2.1 Definição de Variáveis

Podemos imaginar uma variável como o nome de um local onde se pode colocar qualquer valor do conjunto de valores possíveis do tipo básico associado. O nome da variável é um identificador tal como definido anteriormente.

Exemplo



Toda variável deve ser declarada conforme a *sintaxe* abaixo:



São exemplos de declaração de variáveis.

<u>*inteiro*</u> : X1;
<u>*real*</u> : A, B;
<u>*caracter*</u> : FRASE, NOME;
<u>*lógico*</u> : TEM;

A *semântica* de uma declaração de variáveis corresponde à criação de locais na memória rotulada com o nome da variável (identificador) e marcada com o tipo de valores que ela pode conter.

Assim, X1 é o nome de um local de memória que só pode conter valores do tipo *inteiro*. Do mesmo modo, A e B são locais de memória que só podem conter valores *reais*, TEM só pode conter *falso* ou *verdadeiro* e FRASE ou NOME só podem conter conjuntos de *caracteres*.

2.3 COMANDOS BÁSICOS

COMANDO DE ATRIBUIÇÃO. Para a atribuição de um valor a uma variável, usaremos o *símbolo de atribuição* ←, que tem um caráter *imperativo*.

A *sintaxe* do comando é: 

A notação usada para *expressões* é basicamente uma forma linearizada da usada comumente na matemática, tal como ela é conhecida nas linguagens como por exemplo ALGOL [8].

OPERADORES ARITMÉTICOS. Além dos símbolos das quatro operações básicas '+', '−' (binário ou unário), '*', '/' usaremos símbolos para raiz quadrada e exponenciação, por exemplo

$$\sqrt{a} \text{ ou } raiz\,(a),\ (a+b)^n \text{ ou } (a+b)**n$$
$$\sqrt{b^2 - 4*a*c}$$

e os nomes das funções matemáticas comuns, por exemplo

sen (*x*),	*cos* (*x*),	*tg* (*x*), *arctg* (*x*), *exp* (*x*) etc.
abs (*x*)	ou |*x*|,	*sinal* (*x*)
int (*x*)	ou ⌊*x*⌋	parte inteira de *x*: o maior número inteiro que é menor ou igual a *x*
⌈*x*⌉		o menor número inteiro que é maior ou igual a *x*

Usaremos também como nomes de operadores

mod, por exemplo, *m mod i* : resto (módulo) da divisão de *m* por *i*
div ou '÷' por exemplo *n div m* : quociente da divisão inteira de *n* por *m*

O programador tem plena liberdade de introduzir novos operadores ou nomes de funções para adaptar a linguagem às necessidades específicas da sua área de aplicações ou de seu problema, sempre que eles estejam bem definidos, sem deixar margem a ambigüidades.

OPERADORES LÓGICOS. Dentro de relações lógicas, usaremos os conectivos lógicos usuais

e ou ∧ para a conjunção
ou ou ∨ para a disjunção (não exclusiva)
não ou ¬ para a negação

OPERADORES RELACIONAIS. Analogamente, usaremos os conectivos relacionais =, ≠, >, ≥ (ou >=), <, ≤ (ou <=) de significado óbvio.

As prioridades para operações mistas é dada pela tabela a seguir: